UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/894,331	06/28/2001	Anders Hejlsberg	MS180586.01/MSFTP298US	6467

27195 7590 05/30/2008

AMIN. TUROCY & CALVIN, LLP
24TH FLOOR, NATIONAL CITY CENTER
1900 EAST NINTH STREET
CLEVELAND, OH 44114

EXAMINER	
RUTTEN, JAMES D	

ART UNIT	PAPER NUMBER
2192	

NOTIFICATION DATE	DELIVERY MODE
05/30/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com
hholmes@thepatentattorneys.com
lpasterchek@thepatentattorneys.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/894,331
Filing Date: June 28, 2001
Appellant(s): HEJLSBERG ET AL.

Himanshu S. Amin, Reg. No. 40,894
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 3/4/08 appealing from the Office action mailed 7/6/07.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

US 2002/0143521 A1	CALL	10-2002
US 2002/0147745	HOUBEN et al.	10-2002

(9) Grounds of Rejection

The following grounds of rejection are applicable to the appealed claims:

- Claims 1-5, 8, 10-12, 16-20, 22-24, 26, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Call (US 2002/0143521 A1, hereinafter *Call*) in view of Houben et al. (US 2002/0147745, hereinafter, *Houben*).

Claim 1

Call teaches a computer system for parsing XML (see at least FIG.1 & associated text), the system comprising:

- A scanner that parses an XML stream (see at least *DOM interface 145, SAX interface 141* FIG.1 & associated text; *parsing, XML, sequence of nodes, DOM* paragraph [0014]; FIG.5 & associated text) to locate at least one XML token associated with an XML item, the XML stream includes information from at least two data stores (see at least *SAX, character text tokens* paragraph [0072]; paragraphs [0031], [0034], [0044], [0075], [0010]); Further, the disclosure of Call paragraph [0023] discloses data organized at least as complex data structures, such as relational tables, hierarchical object structures etc. (i.e., at least two data stores), .. using special fields called links. Paragraph [0031] further discloses storing each item's physical storage location (i.e., data store) in a lookup table indexed by itemnumber, allowing itemnumber links (i.e., at least two data stores) to be rapidly dereferenced to obtain the location of linked items. Paragraphs [0033]-[0034] also disclose the XML metadata for describing each itemtype (corresponding to each of the

item) comprising a qualified name, i.e., namespace. The same passage explicitly discloses that two items with different names (i.e., namespaces or data stores) are treated as different item types and are assigned different itemtypenumbers. Paragraph [0038] discloses each item is composed of a set of fields. Paragraph [0044] similarly discloses field "Field name" (i.e., XML data) as namespace. Needless to say, it is clear that each item has a corresponding Field name (i.e., namespace identifying the location/data store) in the XML stream that allows the item to be accessed from its data store.

- A reader that selectively pulls the XML item from the XML stream (see at least *DOM, SAX, item/field addressing mechanism* paragraph [0072]); and
- A retriever that retrieves information associated with the pulled XML item (see at least *client program, DOM, XML* paragraph [0072]).

Call does not expressly disclose the retriever exposes data model and/or Infoset information associated with the pulled XML item. However, *Houben* discloses a method of parsing XML (e.g., see *DOM, XML parser* para.[0015]) wherein the retriever exposes data model and/or Infoset information associated with an XML item (e.g., see *DOM, internal data structure 1805, infoset* para.[0055]). *Call* and *Houben* are analogous art because they are both directed to XML parsers. It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Houben* into that of *Call* for the inclusion of exposing the Infoset information. And the motivation for doing so would have been to enable manipulation of the XML document (e.g., adding and deleting nodes and leaf elements of the XML document) by software objects (via exposed methods for operation on the infoset such as add and delete methods) (e.g., see *add method 1815, delete method 1820* para.[0055]).

Claim 2

The rejection of base claim 1 is incorporated. *Call* further teach the XML item is one of a start token, an end token, markup, content, an entity reference, an external reference, an element, a tag, a character data, an attribute, a CDATA section, a comment and a processing instruction (see at least *Element 505, Content 513, Stag 511, Etag 515, Attribute 522, Comment 523, PI 525* FIG. 5 & associated text; *character data, XML* paragraph [0068]).

Claim 3

The rejection of base claim 1 is incorporated. *Call* further teach a checker that determines whether the pulled XML item is well-formed (see at least *well-formed XML documents, XML production rules* paragraph [0073]; *well formed XML documents, XML grammar, production rules* paragraphs [0327]-[0328]; paragraph [0347]; *error messages, failure to parse document, well formed* paragraph [0351]).

Claim 4

The rejection of base claim 1 is incorporated. *Call* further teach a validator that determines whether pulled XML item is valid (see at least *Schema 128, Schema Interpreter 129* FIG.1 & associated text; *XML document, validated, XML schema* paragraph [0073]).

Claim 5

The rejection of base claim 1 is incorporated. *Call* further teach the scanner facilitates navigating a virtual node in a stream of XML nodes (see at least *DOM interface 145, SAX interface 141* FIG.1 & associated text; *parsing, XML, sequence of nodes, DOM* paragraph [0014]; FIG.5 & associated text*; DOM, SAX, item/field addressing mechanism* paragraph [0072]), and resolves an external reference in the XML stream (see at least *URIs, tokenized text strings* paragraph [0075]).

Claim 8

The rejection of base claim 1 is incorporated. *Call* further teach where the reader selectively pulls an XML node from the stream of XML nodes based, at least in part, on data provided to the reader by a parse requestor (see at least *DOM, SAX, item/field addressing mechanism*, *client program, XML* paragraph [0072]).

Claim 10

The rejection of base claim 3 is incorporated. *Call* further teach the checker determines whether the pulled XML item is well-formed base, at least in part, on comparing the pulled XML item to one or more syntax documents (see at least *well-formed XML documents, XML production rules* paragraph [0073]; *well formed XML documents, XML grammar, production rules* paragraphs [0327]-[0328]; paragraph [0347]; *error messages, failure to parse document, well formed* paragraph [0351]).

Claim 11

The rejection of base claim 4 is incorporated. *Call* further teach the validator determines whether the pulled XML item is valid base, at least in part, on comparing the XML item to one or more DTD, schema, and external data representation documents (see at least *Schema 128, Schema Interpreter 129* FIG.1 & associated text; *XML document, validated, XML schema* paragraph [0073]; *XML Schema, DTD* paragraph [0115]).

Claim 12

The rejection of base claim 1 is incorporated. *Call* further teach where at least one of the scanner, the reader and the retriever is an object (see at least *DOM, API* paragraph [0074]).

Claim 16

Call teaches a computer-implemented method for parsing XML, the method comprising:

- Instantiating a pull model parser (see at least *client program, DOM, XML* paragraph [0072]);

Call does not expressly disclose establishing a state (i.e., initial state position) associated with the pull model parser, that is to say, having associated a state machine with the pull model parser. However, these features are deemed to be inherent in the teaching of *Call* because a computing device or a computer where the parser resides is considered to be a state machine associated with the parser wherein each machine instruction [received from the parser code] is input that changes (i.e., updating or repositioning) one or more states (i.e., established initial state position) and may cause other actions/events to take place. Furthermore, each computer's data register stores a state. The ROM from which a boot program is loaded stores a state (the boot program itself is an

initial state). The operating system is itself a state and each application (i.e., parser) that runs begins with some initial state that may change as it begins to handle input (i.e., XML stream). Thus, in view of the forgoing discussion, *Call* clearly teaches

- Establishing a state (i.e., initial state position within the state machine), that is to say, having associated the state machine with the pull model parser (see at least *client program, DOM, XML* paragraph [0072]);
- Accepting a parse request (see at least *client program, DOM, XML* paragraph [0072]);
- Selectively pulling an XML item based, at least in part, on the parse request (see at least *client program, DOM, XML* paragraph [0072]; paragraphs [0031], [0034], [0044], [0075], [0010]); and
- Updating the state based on the selectively pulled XML item (see above discussion).

Call does not expressly disclose the retriever exposes data model and/or Infoset information associated with the pulled XML item. However, *Houben* discloses a method of parsing XML (e.g., see *DOM, XML parser* para.[0015]) wherein the retriever exposes data model and/or Infoset information associated with an XML item (e.g., see *DOM, internal data structure 1805, infoset* para.[0055]). *Call* and *Houben* are analogous art because they are both directed to XML parsers. It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Houben* into that of *Call* for the inclusion of exposing the Infoset information. And the motivation for doing so would have been to enable manipulation of the XML document (e.g., adding and deleting nodes and leaf elements of the XML document) by software objects (via exposed methods for operation on the infoset such as add and delete methods) (e.g., see *add method 1815, delete method 1820* para.[0055]).

Claims 17-20

Claims recite limitations, which have been addressed in claims 3, 10, 4, and 11 respectively, therefore, are rejected for the same reasons as cited in claims 3, 10, 4, and 11.

Claim 22

The rejection of base claim 16 is incorporated. *Call* further teaches where instantiating the pull model parser comprises:

- Associating a stream with the pull model parser (see at least *client program, DOM, XML* paragraph [0072]); and
- Initializing a scanner adapted to facilitate navigating within the stream (see at least *client program, DOM, XML* paragraph [0072]).

Claim 23

The rejection of base claim 16 is incorporated. Claim recites limitations, which have been addressed in claim 16, therefore, is rejected for the same reasons as cited in claim 16.

Claim 24

The rejection of base claim 16 is incorporated. *Call* further teaches where selectively pulling an XML item further comprises:

- Positioning a virtual node over an XML node within a stream of input XML nodes (see at least *client program, DOM, XML, SAX, item/field addressing mechanism* paragraph [0072]); and
- Selectively extracting an XML item from the XML node over which the virtual node is positioned (see at least *client program, XML, DOM, SAX, item/field addressing mechanism* paragraph [0072]); and
- Resolving an external reference in the XML item (see at least *URIs, tokenized text strings* paragraph [0075]).

Claim 26

The rejection of base claim 16 is incorporated. Claim recites limitations, which have been addressed in claim 16, therefore, is rejected for the same reasons as cited in claim 16.

Claim 27

Call teaches a computer readable medium having a tangible component that stores computer executable instructions for a method for parsing XML, the method comprising:

- Operably connecting a pull model parser and a state machine (see *state machine* claim 16);
- Establishing an initial state in the state machine (see *state position* claim 16);
- Accepting a parse request (see claim 16);
- Selectively pulling a first XML item identified in the parse request from a first data store (see claim 16);

- Based at least in part on the first XML item, selectively pulling a second XML item from a second data store (see at least paragraphs [0023], [0031], [0034], [0044], [0075], [0010]).
- Maintaining the state machine in response to one or more events associated with parsing and/or pulling the pulled first and second XML items (see claim 16);
- Checking the pulled first and second XML items to determine whether they are well-formed (see at least *well-formed XML documents, XML production rules* paragraph [0073]; *well formed XML documents, XML grammar, production rules* paragraphs [0327]-[0328]; paragraph [0347]; *error messages, failure to parse document, well formed* paragraph [0351]); and
- Checking the pulled first and second XML items to determine whether they are valid (see at least *Schema 128, Schema Interpreter 129* FIG.1 & associated text; *XML document, validated, XML schema* paragraph [0073]).

Call does not expressly disclose the retriever exposes data model and/or Infoset information associated with the pulled XML item. However, *Houben* discloses a method of parsing XML (e.g., see *DOM, XML parser* para. [0015]) wherein the retriever exposes data model and/or Infoset information associated with an XML item (e.g., see *DOM, internal data structure 1805, infoset* para.[0055]). *Call* and *Houben* are analogous art because they are both directed to XML parsers. It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *Houben* into that of *Call* for the inclusion of exposing the Infoset information. And the motivation for doing so would have been to enable manipulation of the XML document (e.g., adding and deleting nodes and leaf elements of the

XML document) by software objects (via exposed methods for operation on the infoset such as add and delete methods) (e.g., see *add method 1815, delete method 1820* para.[0055]).

(10) Response to Argument

On pages 4-7 (in particular, see bottom of page 6) of Appellants' Brief filed 3/4/08, Appellants essentially argue that the prior art of record, Call, does not disclose " the XML stream includes information from at least two data stores" as claimed in independent claims 1, 16, and 27. This argument was originally addressed on pages 2 and 3 in the 7/6/07 final Office action, which cited Call in support of the rejection.

Paragraphs [0015] and [0017] of the Call reference recite:

> It is a [sic] one object of the present invention to represent character data, particularly natural language text and **markup**, in a more efficient compressed format which requires less storage space and needs less transmission bandwidth, and which can be more rapidly processed than character data. [emphasis added]
> ...
> It is a further object of the invention to store fixed and variable length data as an addressable array of integer values organized to provide efficient data manipulation functions typically performed by hierarchical object oriented data handling systems, including systems conforming to the Document Object Model widely used for storing and manipulating XML and HTML character data

Paragraph [0023] was relied upon in the 7/6/07 Office action and recites:

> In a preferred embodiment of the invention, data stored in the integer array is subdivided into items, and items are subdivided into fields. Items may be organized into more complex data structures, such as **relational tables**, hierarchical object structures, linked lists and trees, and the like, using special fields called links. [emphasis added]

Page 10 lines 10-11 of the originally filed specification provides the following description of a "data store":

> The data stores 250 can include, but are **not limited to**, files, **databases**, pipes, streams, memory and queries. [emphasis added]

The originally filed specification does not appear to particularly limit the definition of the term "data store." As noted above, the disclosure of Call paragraph [0023] discloses data organized at least as complex data structures, such as relational tables, hierarchical object structures etc. (i.e., at least two data stores), .. using special fields called links. Paragraph [0031] further discloses storing each item's physical storage location (i.e., data store) in a lookup table indexed by itemnumber, allowing itemnumber links (i.e., at least two data stores) to be rapidly dereferenced to obtain the location of linked items. Paragraphs [0033]-[0034] also disclose the XML metadata for describing each itemtype (corresponding to each of the item) comprising a qualified name, i.e., namespace. The same passage explicitly discloses that two items with different names (i.e., namespaces or data stores) are treated as different item types and are assigned different itemtypenumbers. Paragraph [0038] discloses each item is composed of a set of fields. Paragraph [0044] similarly discloses field "Field name" (i.e., XML data) as namespace. Needless to say, it is clear that each item has a corresponding Field name (i.e., namespace identifying the location/data store) in the XML stream that allows the item to be accessed from its data store. Thus, Call's markup could be reasonably broadly interpreted as including data from at least two so called “data stores.”

In response to Appellants' argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning (see top of page 7 of Appellants' Brief), it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the Appellants' disclosure, such a reconstruction is proper. See In

re McLaughlin, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971). Further, Appellants point to a perceived lack of disclosure as an indication that such hindsight reasoning has occurred. However, as pointed out above, a reasonable broad interpretation of the claim language allows the reference to read on the claims. Thus, there is no lack of disclosure, and no hindsight reasoning.

At the bottom of page 7 of Appellants' Brief filed 3/4/08, Appellants argue that secondary reference Houben does not disclose a scanner that parses an XML stream to locate at least one XML token associated with an XML item such that the XML stream includes information from at least two data stores. However, as noted on page 4 of the 7/6/07 final Office action, Call, not Houben, was relied upon to disclose these limitations. E.g. see at least Call paragraph [0014] e.g. "parsed into a sequence of nodes" and also paragraph [0073] e.g. "As described in more detail later, the builder 130 imports well-formed XML documents expressed as character data by **parsing that data** in accordance with the XML production rules **into node data** represented by items and fields encoded as integers in an integer array." Further, as noted above, Call discloses information from at least two data stores. Thus, Appellants' argument is not persuasive.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/J. Derek Rutten/
Patent Examiner, Art Unit 2192

Conferees:

Tuan Q. Dam
Supervisory Patent Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192

Wei Zhen
Supervisory Patent Examiner, Art Unit 2191
/Wei Zhen/

Supervisory Patent Examiner, Art Unit 2191